Detecting City Furniture and Vegetation from Mobile Mapping Point Clouds

Martin Kodde Marc van Kreveld Remco Veltkamp

1 Introduction

Governmental organisations such as municipalities and road authorities, have the task to provide a clean and safe public environment to its citizens. As managing the public space becomes increasingly complex, there is a great need for asset management tools that provide up-to-date information about the status of objects, as well as future maintenance cost estimations. These asset management systems rely on up-to-date and detailed topographic information. Such topographic databases should for instance include the location of all traffic signs, lamp posts, trees, etc. Unfortunately, topographic information is not generally available in this level of detail, or the data is not stored in a compatible format.

There are two notable developments in the field of geo-information that will improve the topographic information needed for efficient public asset management. First, there is the development of new information standards for geo-information. With CityGML, a new worldwide standard was set that does not only provide a means to model cities in three dimensions, it also provides classes and attributes for objects commonly found in urban environments [4]. Within the Netherlands, a new data model IMBGT, was derived from CityGML and adopted as a standard [12]. Secondly, the ongoing developments in data acquisition technology provide the opportunity to collect detailed information. The development of Mobile Mapping systems in particular, has resulted in new datasets with huge potential. These datasets are usually stored as point clouds.

However, constructing CityGML city models from point clouds still requires a considerable manual effort. Therefore, we face a need to automate the generation of CityGML models for the use in public asset management. In this work, we focus specifically on detecting city furniture from point clouds.

2 Prior work

Populating CityGML databases with building models has received much research interest over the last years, for example in [14], [10], [8] and [13]. More recently, we have seen research dedicated to city furniture. In [3], [5], [15] and [9] methods are developed to detect city furniture objects from mobile mapping point clouds. There are some clear differences in the approach taken in this prior work. In [5] the detection is performed on the scan lines, while all other methods work on the entire point cloud. Some methods, such as [3] rely on additional data, while others work without any additional input at all. The success rate of the methods as reported by the respective authors varies between 65% and 90%.

3 Objectives

Most city furniture in urban environments includes a part that can be abstracted as a cylindrical object, like traffic signs, street lamps, flag posts, trees, and road bollards. We therefore aim for a method to detect cylindrical parts from mobile mapping point clouds. Given such a point cloud

$\mathcal{P}$ cylindrical objects, we aim to select subsets of points such that all points in a subset lie on the perimeter of the same cylindrical part.

However, objects in the point cloud $\mathcal{P}$ may be occluded, inclined, tapered, or otherwise not be a perfect, vertical cylinder. Similarly, objects such as trees are not perfectly cylindrical. Finally, all points in the point cloud suffer from noise, due to the precision of the scanner and the positioning equipment. To make the detection algorithm robust against these effects, we have postulated the following requirements:

- The algorithm should be able to detect slightly inclined or tapered poles and approximately cylindrical objects
- The algorithm should be robust against shadow resulting from occlusion
- The algorithm should be robust against objects partly surrounded by vegetation
- The algorithm should be robust against objects embedded in larger structures
- The algorithm should be able to detect both small and tall objects

4 Method

4.1 Data pre-processing

As input data we assume a geo-referenced point cloud $\mathcal{P}$ from a mobile mapping system. Often, multiple passes of the mobile mapping system, also called drive-lines in analogy with flight-lines for airborne sensors, are merged into one point cloud. The position of a point $p \in \mathcal{P}$ is known with the precision σ, where σ is defined as the single standard deviation. In most data acquisition systems, we have to differentiate between relative precision σ_{rel} and absolute precision σ_{abs}. The relative precision is the local noise in the point cloud which is predominantly the result of the range accuracy of the laser scanner. The absolute precision describes the precision of the point cloud relative to the coordinate reference system. This error is the result of the platform positioning precision, that is, the precision of the GNSS and inertial positioning. Within a small time span δt this error is systematic. Most high-end Mobile Mapping systems operate with $\sigma_{rel} = 0.01m$ and $0.03m \leq \sigma_{abs} \leq 0.1m$.

These time-dependent systematic effects result in slight offsets of the points between multiple passes of the mobile mapping system. These offsets are typically only a few centimetres large, but this does negatively affect the detection of small objects. We therefore process one pass at a time and merge the results back after the detection process.

As part of the data pre-processing we compute point normals using approximate point indexing by [7] and the method described by [6]. Then we classify all planar surfaces in the point cloud using Efficient Ransac [11]. Since planar surfaces cannot represent cylindrical objects, these surfaces can be removed from the dataset. This does not fundamentally improve the detection rate of the algorithm, but the reduced point cloud size improves the efficiency of the Ransac detection that will be run later.

Finally, if additional data is available, this can be used to our advantage. Building footprints in particular can be used to remove points belonging to buildings from the dataset, as we assume that no city furniture is located very close to a building.

4.2 Detecting cylindrical objects

Several methods exist to detect cylinders from point clouds. Both the Hough Transform [1] and Ransac [2], in particular Efficient Ransac [11], are known to provide detection of cylinders from point clouds. The objects we are looking for are locally cylindrical. We therefore subdivide the point cloud in multiple horizontal slices. Using these slices allows us to detect objects with a locally cylindrical shape. This helps in particular for objects that are inclined or tapered. The slice width is a function of point density and the objects to detect. Slices are processed bottom-up. When processing a slice we look if we have prior information about cylindrical objects. This prior information may result from previous slices below the current slice, or alternatively from additional data. If such prior information is available, we test if the cylindrical object is also present in the current slice. The next step is to detect new cylinders in the current slice. City furniture objects obviously cannot start in the air without a connection to the ground, but it may be that the object was occluded at lower height. To detect new cylinders, we run an adapted version of Ransac over the point cloud slice.

Ransac is a method to detect a parametrised model from a set of points, by randomly selecting a minimal subset $S \in \mathcal{P}$ and testing if the model defined by this subset has sufficient support from the other points in the set. This is done multiple times, and the model with the largest support is accepted. Then the points supporting the model are removed. This method is repeated to find more models. A cylinder requires a subset of three points in a minimal subset, because three points uniquely determine a vertical cylinder but two points do not. We apply a type of guided sampling to increase the probability that a triple of randomly selected points will create cylinder with high enough support. Contrary to typical Ransac applications, the number of points that support a model is small, typically only 10 to 20 for a traffic sign pole in a single slice. In addition, most cylinders are only visible in the data from one side, and might suffer from additional occlusion by other objects. When Ransac is applied with such a low candidate count threshold, a large number of false positives will be detected. During the Ransac detection we therefore enforce the following constraints:

- When two cylinders intersect, the candidate sets of both cylinders should also create a valid cylinder
- No points are allowed in the interior of the cylinder
- The number of points surrounding the cylinder should be below a given threshold

4.3 Results and conclusions

We have tested this algorithm on a mobile mapping point cloud in the city of Leiden. We found that most of the cylindrical city furniture objects can be detected correctly, but additional work is needed to reduce the false positives.

We aim to further improve the approach and to add classification of the detected cylindrical objects into CityGML classes. This will result in a method to populate the CityGML city furniture classes automatically from mobile mapping point clouds.

References

[1] D. Ballard. Generalizing the Hough Transform to Detect Arbitrary Shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[2] M. A. Fischler and R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6), 1981.

[3] A. Golovinskiy, V. G. Kim, and T. Funkhouser. Shape-based recognition of 3d point clouds in urban environments. *Computer Vision*, 2009.

[4] G. Gröger, T. H. Kolbe, C. Nagel, and K.-H. Häfele. OpenGIS City Geography Markup Language (CityGML) Encoding Standard. Technical report, Open Geospatial Consortium Inc., 2011.

[5] M. Lehtomäki, A. Jaakkola, J. Hyyppä, A. Kukko, and H. Kaartinen. Detection of Vertical Pole-Like Objects in a Road Environment Using Vehicle-Based Laser Scanning Data. *Remote Sensing*, 2(3):641–664, Feb. 2010.

[6] N. J. Mitra, A. Nguyen, and L. Guibas. Estimating Surface Normals in Noisy Point Cloud Data. *International Journal of Computational Geometry & Applications*, 14:261—-276, 2004.

[7] M. Muja and D. G. Lowe. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. *International Conference on Computer Vision Theory and Applications (VISAPP'09)*, 2009.

[8] S. Oude Elberink and G. Vosselman. 3D information extraction from laser point clouds covering complex road junctions. *The Photogrammetric Record*, 24(March):23–36, 2009.

[9] S. Pu, M. Rutzinger, G. Vosselman, and S. Oude Elberink. Recognizing basic structures from mobile laser scanning data for road inventory studies. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(6):S28–S39, Dec. 2011.

[10] F. Rottensteiner and C. Briese. Automatic generation of building models from LIDAR data and the integration of aerial images. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences of the ISPRS*, 34(3/W13):174–180, 2003.

[11] R. Schnabel, R. Wahl, and R. Klein. Efficient RANSAC for Point-Cloud Shape Detection. In *Computer Graphics Forum*, volume 26, pages 214–226. Wiley Online Library, 2007.

[12] L. van den Brink, J. Stoter, and S. Zlatanova. Modeling an application domain extension of citygml in uml. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVIII:14–17, 2012.

[13] T. van Lankveld, M. van Kreveld, and R. Veltkamp. Identifying rectangles in laser range data for urban scene reconstruction. *Computers & Graphics*, 2011.

[14] G. Vosselman and S. Dijkman. 3D building model reconstruction from point clouds and ground plans. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 34(3/W4):37–44, 2001.

[15] H. Yokoyama, H. Date, S. Kanai, and H. Takeda. Pole-Like Objects Recognition from Mobile Laser Scanning Data using Smoothing and Principal Component Analysis. In *ISPRS Workshop, Laser scanning*, number Lehtomaki, 2011.